

Parallel Streaming for a Multi-homed Dash client

Ali Hodroj, Marc Ibrahim
Saint Joseph University of Beirut
Faculty of Engineering, ESIB
Mar Roukos, Lebanon
ali.hodroj1@net.usj.edu.lb
marc.ibrahim@usj.edu.lb

Yassine Hadjadj-Aoul
Univ Rennes, Inria, CNRS, IRISA
Rennes, France
yassine.hadjadj-aoul@irisa.fr

Abstract—Nowadays, multimedia streaming traffic reaches 71 % from the mobile data traffic over the world and most of the multimedia services use Dynamic adaptive streaming over HTTP (DASH) to adjust video delivery to the dynamic network environment and achieve higher user Quality of Experience (QoE) levels. Moreover, 90 % of the video traffic is consumed by smart devices equipped with multiple network interfaces (Wifi, 3G, and 4G) known as multi-homed devices. In this paper, we propose a new DASH-parallel streaming solution, which exploits the diversity of access network connections to improve video quality over DASH protocol. Experimental Results show that our proposed approach improves the perceived QoE in different network conditions, without requiring further energy expenditure.

Index Terms—MPEG-DASH, multi-homing and parallel streaming.

I. INTRODUCTION

According to the Cisco Visual Networking Index, the demand for video traffic is increasing and exceeds 78% of the total mobile data traffic by 2021 [5]. Most of the video services are today based on adaptive streaming techniques that allow a video to be played based on the end user's network conditions. Therefore, the quality of the video and the audio streams are automatically adjusted accordingly, enhancing the user's experience [15].

Dynamic adaptive streaming over HTTP (DASH), which is becoming the dominant protocol for video delivery, works by dividing the media content into a sequence of small HTTP-based file segments, each chunk consists of a short interval of playback time of the media content [14]. In DASH, video content is represented by two elements: The media Presentation Description (MPD) file and the video/audio Segments [3]. The MPD is an XML manifest file that identifies multiple components and the location of all alternative streams. After analyzing the MPD, the client requests and download an appropriate segment, which guarantees smooth playback of the video file and enhances the QoS and performance since the client knows its capabilities (i.e. the received throughput, the delay, the terminal's potential – screen resolution –) [10] [11] [19].

From a traffic perspective, smartphones will continue to dominate mobile traffic (up to 93% by 2022). As smartphones are multi-homed devices [1], usually equipped with multiple radio interfaces (3G, 4G, and WiFi), this potentially allows the device to connect to multiple networks at the same time.

This emerging diversity of access networks could be used to improve the perceived video quality.

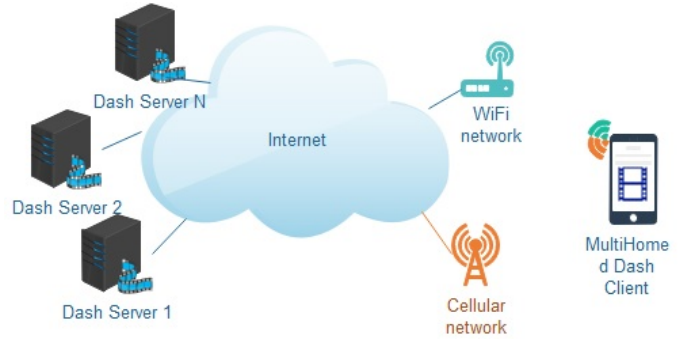


Fig. 1. Multihoming Infrastructure

In this paper, we study the use case of a multi-homed DASH client that aims to watch a video content replicated in many DASH servers, as shown in Figure 1. For this purpose, a parallel streaming mechanism over DASH is proposed to enhance the QoE for multi-homed clients. Our solution permits the client to request a segment simultaneously using both available connections, to provide a kind of bandwidth aggregation mechanism that maximizes the available resources (connections), which allow improving the received video-rate without any quality degradation due to the variation of the network's conditions on both connections. Additionally, it provides the following features:

- More robustness using the content replication to ensure high data availability, as a user can use multiple sources to prevent any quality degradation due to congestion at the server side.
- Parallel streaming over DASH, as a Dash client can request a segment using multiple connections simultaneously.
- Smart downloading mechanism. Indeed, the main question when using multi-homing is how to ensure that the way of splitting the data among multiple connections is efficient? our downloader response about this question is described in Section 3.
- An energy-saving mechanism, as using parallel streaming increase power consumption and without any quality improvement in some cases.

The remainder of the paper is organized as follows. Sections II describes the latest solution proposed in Multi-homing over DASH. Section III presents a description of our proposed solution. we evaluate our Parallel Streaming algorithm over DASH, in section IV. Finally, a conclusion and future research ideas are provided in Section V.

II. RELATED WORK

This section represents a summarized survey about the related works in multi-homing and multi-paths solutions for video streaming. Min Xing formulate the video streaming process as Markov decision process (MDP) and propose a multi-link rate adaption (MLRA) mechanism using multihoming features [18]. At each set of segments request (considered as time step), we use MDP to optimize the Quality of Experience of video content by evaluate the current situation and reward function of available states. The objective of this approach is to improve the QoE while reducing the cost of using multiple connections. MLRA algorithm selects the next segment based on the current buffer level and the estimated available bandwidth. It takes into consideration the calculated reward function of each decision (using only a wifi link or both links or smoothing).

Using the same concept of MDP problem formulation, [17] proposes a real-time adaptive search (RTRA) algorithm to improve video performances. The results show that RTRA achieves higher quality in different bandwidths scenarios (rapid and slow-changing) and reduce startup delay, which leads to user stop watching the video in most cases, up to half and better smoothness.

A cross-layer approach was proposed in [6] based on an interaction between the application and transport layer which lead to decrease video distortion and improve video quality. This approach consists of using Multi-Path Transmission Control Protocol (MPTCP) as the transport layer with some additional features considered to add feedback that indicates the path to be selected by MPTCP for the next packet. Also, an adaptive content-aware scheduler is used at the application layer to estimate the playback deadline and thus determine which video segment and when is given for transmission. The objective of this solution is to maximize the received data on time, to prevent the receiving buffer from being fully occupied by many packets waiting for the delayed packet (head-of-line blocking). Also using the content awareness, it ignores the transmission of segments that will miss their playback deadline when checking the Picture Order Counts (POC) of each segment and gives more priority to the segments which their deadline period is near to expire. Another cross-layer approach was proposed in [13] using DASH and MPTCP. A Distributed Decision Engine (DDE) was proposed as a network monitor systems responsible for providing the accurate network information's on each interface. This network information's are used in the Dash bit rate adaption, then MPTCP is responsible for sending the data over multiple paths. The results show that more accurate information's lead to better bit rate adaption and improve video performance.

Similar to before, the MP-DASH framework [7] was proposed to extend MPTCP for dealing with multiple network interfaces over DASH protocol. Two modules were integrated for this purpose the scheduler and video adapter. The scheduler is based on user input preferences about network paths and throughput aggregation between paths. It consists of two parts: decision function that determines how to manage paths and enforcement function that operates the decisions. The video adapter is implemented to enhance the Dash adaptive bit-rate mechanism by exchanging the useful information (throughput, segment size and deadline time) between the scheduler and player. The main idea from this work is to meet user quality requirements with the Wifi link alone and use the cellular data (3G) when wifi cannot deliver data before the deadline period. The result, in comparison to default MPTCP, shows a valued decrease in cellular usage up to 99% and 85% in energy consumption. A new cross-layer approach between transport and network layer was proposed in [12]. In this work, MPTCP is used with a path controller module based on Software Defined Networking (SDN) platform. Network information's about different paths is monitored using the SDN controller and periodically transmitted to SDN clients permitting to evaluate and characterize the available paths (remove paths that lead to out-of-order delivered packets due to low throughput). This solution was implemented over DASH and results show better QoE is achieved with this path controller module on MPTCP.

The author in [4] modified the DASH protocol to support multihoming features. This modification consists of adding a scheduler, adaptation algorithm and bandwidth estimator on the DASH plugin. The bandwidth estimator measures the available bandwidth on each interface by sniffing packets and estimates the bandwidth in the future based on present and previous measurements. The adaption algorithm uses the estimated bandwidth for rate adaptation for each requested segment. The scheduler is responsible for scheduling requests for segments.

A video delivery utilizing over multiple paths was proposed in [9] to achieve low latency in DASH video streaming. Two adapted mechanisms are proposed in this approach. The first determines the best byte-range size of each divided segment over multiple paths, in order to receive them before the deadline. Thus, the larger byte ranges are transmitted over better paths which decrease the delay and solve the HOL problem. The second one is responsible for selecting a suitable representation with the highest quality that can arrive on time.

A network selection mechanism based on the Multi-Armed bandit algorithm was proposed in [8] for a DASH multihomed client. The proposed solution permit the user to select the suitable interface at each decision time, which enhance the average perceived video quality. Also the default adaptive bit-rate rule of DASH require some modifications to deal with different network conditions over different paths, which was validated in the results of several experiments with and without this modification.

III. PROPOSED SOLUTION

In this paper, we propose a Parallel Streaming solution over multi-homed DASH (PS-DASH) for video streaming as an application solution that does not require any kernel modifications at the server or the client sides.

Our solution benefits are:

- **Channel diversity:** Using multiple connections allows the user switching to an alternate path in case of failure or congestion in the current path. Moreover, it permits the user to use both connections simultaneously, a kind of bandwidth aggregation, to achieve higher video quality. Many challenges may occur when adding multi-path features to video transmission starting from Head Of Line blocking (HOL) problem, network heterogeneity, out-of-orders packets, and end-to-end delay. Our solution deals with such challenges simply and smartly, as described in the next section.
- **Source diversity:** The increased demand, for high-quality video services, was observed in the CISCO forecast. This may lead to an overload at the server-side, which results in increasing the end-to-end latency and quality degradation at the client-side. To deal with such a problem, we take the advantages of Content Delivery Networks (CDN) where the content is replicated at multiple servers' locations and permit the client to watch the video from several sources. Thus, each client is able to download the video content from various sources based on the expected high quality that can be achieved from each server [2].
- **Quality enhancement:** Using parallel streaming and source diversity permit the client to increase its network speed. One of the main challenges in parallel streaming is the respect of the packets' order in video transmission to avoid packets' jitter, unnecessary waiting time in the buffer causing buffer overflow or HOL problems. Moreover, in adaptive streaming, the main advantage is to choose the corresponding video quality based on network conditions. Our proposed solution keeps allowing the adaptive bitrate algorithm to more accurate bitrate adaptive decisions due to a more accurate measurement of network conditions over multiple connections.
- **Energy savings:** Our solution takes into consideration the use of parallel streaming when it leads to enhanced quality, which reduces energy consumption. Our solution keeps monitoring the network conditions and checks the available bit-rate of video content. Thus, if one connection is fast enough to meet the higher available video quality, no need to use a second connection.

A. Model

- $\mathcal{Q} = \{q_j\}_{j=1..N_Q}$ is the set of available N_Q video qualities.
- $\mathcal{I} = \{I_i\}_{i=1..N_I}$: the set of available N_I network interfaces. Only two interfaces are considered in this work.
- P_s : the segment video duration.

- t : the time step for the PS-DASH algorithm. The duration of the time step is equal to $n \times P_s$, where n is the fixed number of segments to download in a time step.
- $S(t)$: the segment size at time t , which is equal to $\sum_{i=1}^{N_I} s_i(t)$, where $s_i(t)$ is the amount of data from segment s download through interface i at time t .
- $b(t)$: the amount of buffered video data at time t .
- $T(t)$: the time required to download the segment $S(t)$ at time t .
- $w(t)$: the throughput to download segment $S(t)$ at time t , which is equal $\lceil \frac{S(t)}{T(t)} \rceil$ equal to $\sum_{i=1}^{N_I} w_i(t)$, where $w_i(t)$ is the throughput of interface i at time t .

B. PS-DASH Design

Parallel streaming over the Dash algorithm is based on using multiple connections simultaneously to download each requested chunk from the client in a fastest way.

Many studies have been developed to determine how we divide different segments or split a segment among multiple connections. The splitting mechanism was a challenge because it leads to quality degradation when network conditions are wrongly predicted or estimated. For this purpose, we use a simple way, which doesn't need any network estimation and results in a fastest download. The idea here is to start downloading the current segment from the beginning of the segment, via the first connection, while downloading it from the end on the other interface (see Fig. 2). The download stops when the entire segment has been downloaded, as described by the MPD file.



Fig. 2. Smart Downloader

The main concept of DASH is the adaptive bitrate algorithm that permits the user to switch between the available video rates according to its network conditions and other information that affects the video performances (e.g. like the buffer level). Using multi-homing affects these rules and leads in some cases to quality degradation. Indeed, the decision on the choice of the segment's quality can be based on the conditions of another interface, with a very different status. Our smart segment downloader doesn't suffer from this problem as it is working in intra-segment, which means that it doesn't affect the measurements of the available bitrate (ABR). Thus, it is based on the whole segment not a part of the segment.

In order to estimate the network conditions, we propose to use the average throughput \bar{w}_i to download the next segment.

$$\bar{w}_i(t+1) = \alpha \bar{w}_i(t) + (1 - \alpha) w_i(t) \quad (1)$$

Where α represents a weighting factor giving more or less weight to the current measured throughput.

Also, another use case of switching to single connection streaming instead of parallel is when the current receiver buffer is full, thus, no need for a faster transmission.

Algorithm 1 PS-DASH Algorithm

```

1:  $q_{init} = \text{DEFAULT}$ 
2:  $t = 0$ 
3:  $[S], w_0, w_1 = \text{REQUEST}(n, q_{init}, 2)$   $\triangleright n$  is the number of
   requested segments
4: Push  $[S]$  to DASH player
5: while  $t < \frac{c}{c}$  do  $\triangleright$  while video not finished
6:    $\bar{w}_0(t+1) = \alpha \bar{w}_0(t) + (1 - \alpha) w_0(t)$ 
7:    $\bar{w}_1(t+1) = \alpha \bar{w}_1(t) + (1 - \alpha) w_1(t)$ 
8:   if  $(Q_{max} > \max_{\bar{w}_0, \bar{w}_1})$  AND  $(b(t) \neq \text{FULL})$  then
9:      $i = 2$ 
10:  else if  $\bar{w}_0 > \bar{w}_1$  then
11:     $i = 0$ 
12:  else
13:     $i = 1$ 
14:   $[S], w_0, w_1 = \text{REQUEST}(n, Q, i)$ 
15:   $t = t + 1$ 
16: function REQUEST( $n, Q, i$ )
17:  if  $i == 0$  then
18:    Download segments via interface 1
19:  else if  $i == 1$  then
20:    Download segments via interface 2
21:  else
22:    Download segments via both interfaces
23:  return  $[S], w_0, w_1 -$ 

```

As mentioned in Algorithm 1, the Dash client starts downloading n segments through both interfaces, using the function $\text{Request}(n, q_{init}, 2)$ where q_{init} is the default quality request based on DASH ABR, using the Smart Downloader mechanism. After receiving the n segments, the function measures the throughput of each interface i , which is equal to the amount of data S_i downloaded through this interface divided by the downloading time. Then, the algorithm estimates the throughput of each interface based on the exponential weighted moving average, as shown in Eq. (1). Before requesting a new set of segments, the algorithm checks if the maximum available quality (already known from the MPD file) can be reached through one of the two interfaces alone, and keep track of the current buffer level. If one of these two conditions is satisfied, the algorithm decides to request the next set of segments using single transmission and through the highest available throughput. Otherwise, it uses the parallel transmission mechanism with the same procedures until the end of the video.

IV. PERFORMANCE EVALUATION

We implemented a test-bed (see Figure 3) to evaluate our proposed solution in different scenarios, taking into consideration different network connections characteristics. Our testbed consists of a DASH client with multiple interfaces, each interface is connected to a DASH server. We use the WonderShaper application on each server to control the network characteristics and emulate the required variations [16]. We focus, in our measurements, on the following parameters to evaluate our solution:

- 1) **Average video bitrate:** the average of received video quality.
- 2) **Buffering time:** the time of empty buffer occurrences waiting to receive the segments for streaming.
- 3) **Gain parameter:** the quality enhancement resulting from using parallel streaming.
- 4) **Energy Saver:** prevent using parallel streaming, which causes more energy consumption, when there is no possible quality enhancement.

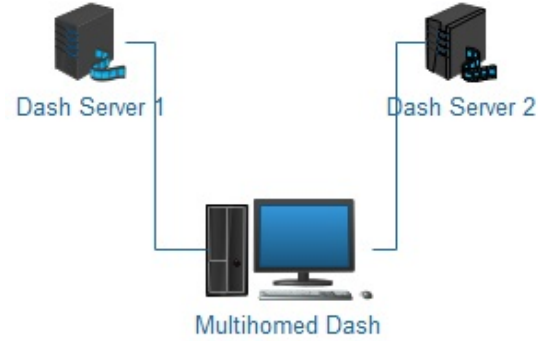


Fig. 3. Experiment Lab Testbed Setup

A. Reference Measurement

we use, in this experiment, a video content with 8 available qualities (760,1013,1255,1884,3134,4953,915,14932 Kbps). we evaluate the performance of the DASH protocol, as reference measurements, with different network speeds (1,2,3,4,5,10,15,20 Mbps) using a single transmission (single interface).

The scenario consists of a standard DASH Client that watches the video content. Each scenario is repeated with different network throughput's in three types of network variations: without, smooth and rough variation. the smooth variation refer to speed variation with $\pm 20\%$ each 2 minutes, while the rough variation refer to speed variation with $\pm 40\%$ each 1 minutes.

The result, in Figure 4, shows how the different network throughput affect the average received video quality in DASH. Better quality is achieved in fast throughput with stable network conditions, and it decreases according to throughput decreases.

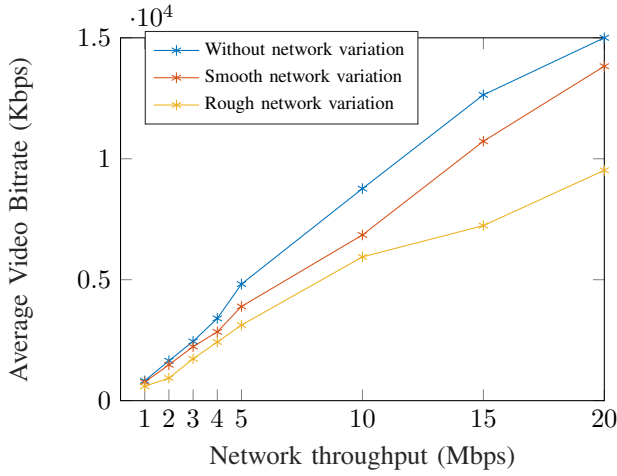


Fig. 4. Video quality variation in function of Throughput variations

Figure 5 shows the variation of buffering time (occurrence and duration) in function of throughput speed variation for different network characteristics. Stable network refers to better video performance, while rough network variations result in more empty buffer occurrences with different duration.

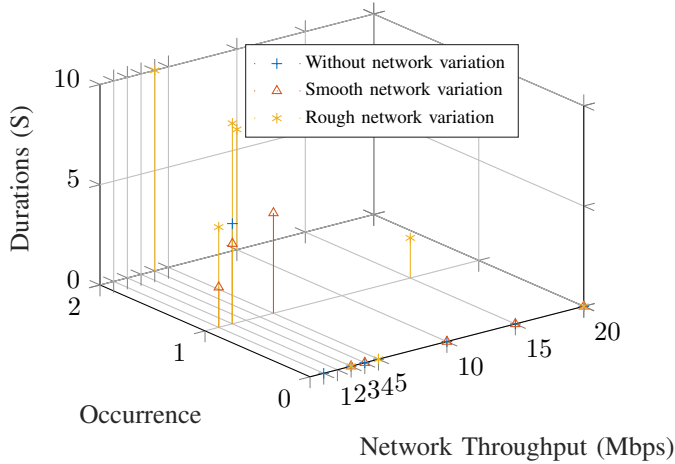


Fig. 5. Video quality variation in function of network speed variation

B. PS-DASH Measurement

We divide our experiment of evaluation into two parts. In Part 1, we study quality enhancement as a metric of evaluation, while in part 2, we study the cost of using parallel streaming in terms of energy consumption.

1) *Quality Evaluation*: In these experiments (Figure 6), we refer to Multi-homed DASH client that watches the same video content using both connections simultaneously with a combination of different network throughput. Then each scenario is repeated with the three types of network characteristics (without, smooth and rough intra-variation of network characteristics).

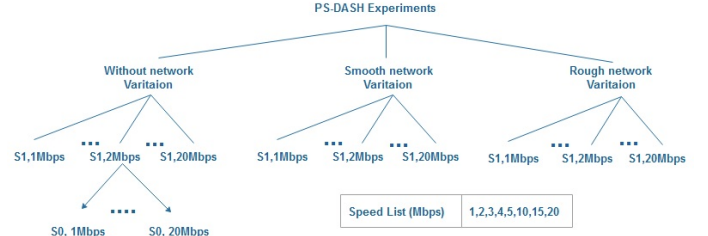


Fig. 6. PS-DASH Experiments

To evaluate our algorithm, 192 scenarios were run: Each scenario consists of a DASH Multi-homed client watching the video content, while fixing the speed of the first connection and changing the speed of the second connection using all speed values, from the speed list. The 64 scenarios, resulting from the combination of the speed list, are repeated with the three types of network variations. We calculate the Enhancement E (Equation 2) resulting from comparing the average video bitrate Q_2 of our algorithm to that one measured from using the highest throughput alone Q_1 in a single transmission.

$$E = \frac{Q_2 - Q_1}{Q_2} \quad (2)$$

Figure 7 shows the Cumulative Distributed Function (CDF) of the average enhancement calculated from these sets of experiments, when comparing our algorithm using the Parallel Streaming over DASH to the standard Dash (single transmission) using the highest connection. The result shows quality enhancement between 10% to 95% in 80% from the scenarios, while 10% from the 192 scenarios were subject to quality degradation in comparison to single transmission. Besides, we can notice the absence of any empty buffer occurrences when using the PS-DASH algorithm.

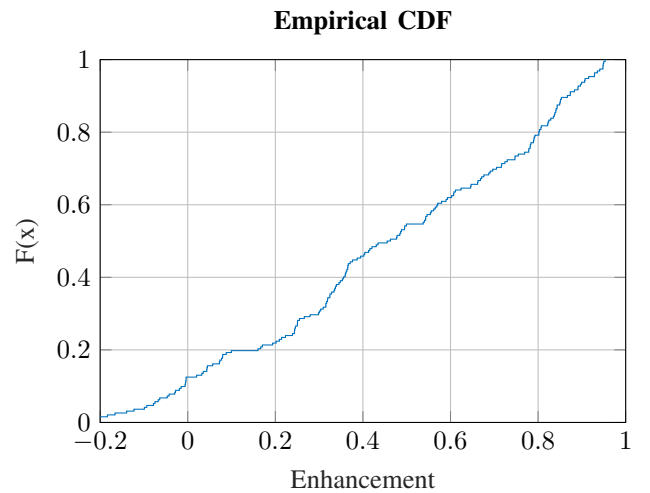


Fig. 7. Average Bitrate Enhancement of MP-DASH Algorithm

2) *Energy Consumption Evaluation*: In these measurements, we refer to a Multi-homed client that watch different

video content, each video content has different bitrate representations. Each scenario is repeated two or three times using different speeds (as shown in Table I). The objective of these measurements is to check the network utilization in different scenarios, to ensure that our algorithm doesn't use the parallel streaming when the maximum quality is reached through the speed of the highest single connection.

Video content	Video qualities (Kbps)	Speed scenarios (Mbps)
1	760;1013	[1;1],[1;2]
2	760;1013;1255	[1;2],[1;3]
3	760;1013;1255;1884	[1;2],[1;3]
4	760;1013;1255;1884;3134	[1;2],[1;3],[1;4]
5	760;1013;1255;1884;3134;4953	[1;3],[1;4],[1;6]

TABLE I
ENERGY CONSUMPTION MEASUREMENT SET

Figure 8 represents the average video bitrate received for different video contents using different sets of network speeds. The Parallel Streaming usage is mentioned on top of each result. One can observe that the parallel streaming usage is decreasing in all cases when the highest available quality can be achieved through one network connection. For example, in the first video content, we have two available qualities, and we make two measurements that refer to two different sets of throughput. The result shows that parallel streaming usage is decreased from 70% to 18% while the highest video quality (1013 kbps) can be achieved through the second connection throughput (2Mbps) alone. In all scenarios, we observe the same conclusion with a different variation of decreasing.

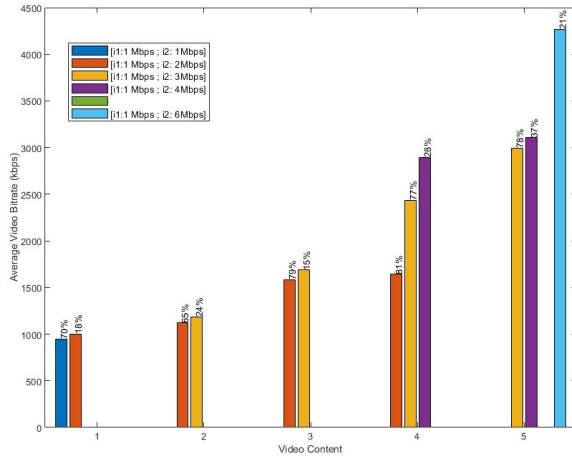


Fig. 8. Parallel Streaming usage in function on speed variations for different video contents

V. CONCLUSION

In this paper, we propose a Parallel Streaming algorithm for multi-homed DASH clients. We showed that the PS-DASH algorithm improves the average video bitrate while

reduces rebuffering time. Furthermore, our algorithm takes into consideration the potential wastage of energy when using parallel streaming. We are planning to upgrade our algorithm to take into consideration the current buffer status related to the rate of decoding to reduce energy consumption. Moreover, we are planning to study the network capabilities of multiple connections, comparing to available video quality bitrate, not just checking if the highest qualities can be archived using single streaming.

REFERENCES

- [1] Samsung galaxy s5 download booster, 2019. "https://galaxys5guide.com/samsung-galaxy-s5-features-explained/galaxy-s5-download-booster/".
- [2] V. K. Adhikari, Yang Guo, Fang Hao, M. Varvello, V. Hilt, M. Steiner, and Z. Zhang. Unreeling netflix: Understanding and improving multi-cdn movie delivery. In *2012 Proceedings IEEE INFOCOM*, pages 1620–1628, March 2012.
- [3] T. Casey and G. Muntean. Mpeg-dash-based framework for improving end-user video experience in heterogeneous multi-network wireless environments. In *2016 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, pages 1–6, June 2016.
- [4] Y. Chowrikoppalu and P. Gowda. *Multipath Adaptive Video Streaming over Multipath TCP*. PhD thesis, University of Saarland, 2013.
- [5] Cisco. Cisco visual networking index forecast and methodology, 2019. Available at <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-forecast-qa.html>.
- [6] Xavier Corbillon, Ramon Aparicio Pardo, Nicolas Kuhn, Géraldine Texier, and Gwendal Simon. Cross-layer scheduler for video streaming over mptcp. pages 1–12, 05 2016.
- [7] Bo Han, Feng Qian, Lusheng Ji, and Vijay Gopalakrishnan. Mp-dash: Adaptive video streaming over preference-aware multipath. pages 129–143, 11 2016.
- [8] A. Hodroj, M. Ibrahim, Y. Hadjadj-Aoul, and B. Sericola. Enhancing dynamic adaptive streaming over http for multi-homed users using a multi-armed bandit algorithm. In *2019 15th International Wireless Communications Mobile Computing Conference (IWCMC)*, pages 692–697, June 2019.
- [9] P. Houzé, E. Mory, G. Texier, and G. Simon. Applicative-layer multipath for low-latency adaptive live streaming. In *2016 IEEE International Conference on Communications (ICC)*, pages 1–7, May 2016.
- [10] Dynamic adaptive streaming over http (dash) — part 6: Dash with server push and websockets. Standard, International Organization for Standardization, March 2017.
- [11] C. Mueller, S. Lederer, C. Timmerer, and H. Hellwagner. Dynamic adaptive streaming over http/2.0. In *2013 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, July 2013.
- [12] H. Nam, D. Calin, and H. Schulzrinne. Towards dynamic mptcp path control using sdn. In *2016 IEEE NetSoft Conference and Workshops (NetSoft)*, pages 286–294, June 2016.
- [13] T. Ojanperä and J. Vehkaperä. Network-assisted multipath dash using the distributed decision engine. In *2016 International Conference on Computing, Networking and Communications (ICNC)*, pages 1–6, Feb 2016.
- [14] I. Sodagar. The mpeg-dash standard for multimedia streaming over the internet. *IEEE MultiMedia*, 18(4):62–67, April 2011.
- [15] J. Vlaović, S. Rimac-Drlje, F. Vranješ, and R. P. Kovač. Evaluation of adaptive bitrate selection algorithms for mpeg dash. In *2019 International Symposium ELMAR*, pages 73–76, Sep. 2019.
- [16] Wondershaper, 2011. Available at <https://github.com/magnifico/wondershaper>.
- [17] M. Xing, S. Xiang, and L. Cai. A real-time adaptive algorithm for video streaming over multiple wireless access networks. *IEEE Journal on Selected Areas in Communications*, 32(4):795–805, April 2014.
- [18] Min Xing, Siyuan Xiang, and Lin Cai. Rate adaptation strategy for video streaming over multiple wireless access networks. pages 5745–5750, 12 2012.
- [19] Y. Zhou, Y. Duan, J. Sun, and Z. Guo. Towards simple and smooth rate adaption for vbr video in dash. In *2014 IEEE Visual Communications and Image Processing Conference*, pages 9–12, Dec 2014.